

IN THE CLAIMS

Please amend the claims as follows:

1. **(Previously Presented)** A method of detecting a computer virus, comprising:
emulating computer executable code in a subject file;
detecting at least one modification to a memory state of a computer system, wherein the
at least one modification:
 is caused by the emulation of the computer executable code; and
 comprises insertion of a pointer to a viral exception handler, the pointer associated
with a particular exception;
and
detecting at least one instruction, wherein the at least one instruction forces the particular
exception.
2. **(Previously Presented)** The method of Claim 1, wherein:
the at least one modification further comprises installation of the viral exception handler.
3. **(Previously Presented)** The method of Claim 1, wherein the particular exception
comprises at least one of the following:
a divide-by-zero arithmetic operation;
an execution of an undefined computer instruction; and
a memory access to an undefined or illegal memory address.
4. **(Canceled)**

5. **(Previously Presented)** A method of detecting a computer virus, comprising:
emulating computer executable code in a subject file;
detecting at least one modification to a memory state of a computer system, wherein:
the memory state comprises a particular interrupt associated with a legitimate interrupt handler; and
the at least one modification:
is caused by the emulation of the computer executable code;
comprises installation of a viral interrupt handler; and
associates the particular interrupt with the viral interrupt handler instead of the legitimate interrupt handler;
and
detecting at least one instruction, wherein the at least one instruction forces the particular interrupt.
6. **(Previously Presented)** The method of Claim 5, further comprising:
detecting writing of a pointer to at least one predetermined address in a system memory for storing an interrupt handler pointer.
7. **(Previously Presented)** The method of Claim 5, further comprising:
detecting use of a predetermined instruction to retrieve an address in a system memory corresponding to an interrupt descriptor table.

8. **(Currently Amended)** A computer program for detecting a computer virus, the program embodied on a computer-readable medium, that when executed causes a computer to:

- emulate computer executable code in a subject file;
- detect at least one modification to a memory state of a computer system, wherein the at least one modification:
 - is caused by the emulation of the computer ~~[-]~~ executable code; and
 - comprises insertion of a pointer to a viral exception handler, the pointer associated with a particular exception; and
- detect at least one instruction, wherein the at least ~~on~~ **one** instruction forces the partiular exception.

9. **(Currently Amended)** A computer program for detecting a computer virus, the program embodied on a computer-readable medium, that when executed causes a computer to:

- emulate computer executable code in a subject file;
- detect at least one modification to a memory state of a computer system, wherein [[:]] :
 - the memory state comprises a particular interrupt associated with a legitimate interrupt handler; and
 - the at least one modification:
 - is caused by the emulation of the computer executable code;
 - comprises installation of a viral interrupt handler; and
 - associates the particular interrupt with the viral interrupt handler instead of the legitimate interrupt handler; and
- detect at least one instruction, wherein the at least one instruction forces the particular interrupt.

10. **(Canceled)**

11. **(Canceled)**

12. **(Canceled)**

13. **(Previously Presented)** An apparatus for detecting computer viruses, comprising:
an emulator component operable to emulate computer executable code in a subject file;
and

a detector component operable to:

detect at least one modification to a memory state of a computer system, wherein
the at least one modification:

is caused by emulation of the computer executable code; and
comprises installation of a viral exception handler;

and

detect at least one instruction, wherein the at least one instruction forces a
particular exception associated with the viral exception handler.

14. **(Previously Presented)** The apparatus of Claim 13, wherein the particular
exception comprises at least one of the following:

a divide-by-zero arithmetic operation;
a memory access to an undefined or illegal memory address; and
execution of an undefined computer instruction.

15. **(Previously Presented)** The apparatus of Claim 13, wherein the at least one
modification further comprises writing of a pointer to the viral exception handler, the pointer
associated with the particular exception.

16. **(Previously Presented)** An apparatus for detecting computer viruses, comprising:
an emulator component operable to emulate computer executable code in a subject file;
and

a detector component operable to:

detect at least one modification to a memory state of a computer system, wherein
the at least one modification:

is caused by emulation of the computer executable code; and
comprises installation of a viral interrupt handler;

and

detect at least one instruction, wherein the at least one instruction forces a
particular interrupt associated with the viral interrupt handler.

17. **(Canceled)**

18. **(Previously Presented)** The apparatus of Claim 16, wherein the at least one
modification further comprises writing of a pointer to the viral interrupt handler, the pointer
associated with the particular interrupt.

19. **(Previously Presented)** The apparatus of Claim 16, wherein the at least one
modification further comprises use of a predetermined instruction to retrieve an address in a
system memory corresponding to an interrupt descriptor table.

20. **(Previously Presented)** The method of Claim 1, wherein the computer system
comprises a first memory component and a second memory component, and wherein access to
the second memory component is more restricted than access to the first memory component.

21. **(Previously Presented)** The method of Claim 20, wherein the viral exception
handler attempts to modify the second memory component.

22. **(Previously Presented)** The method of Claim 5, wherein the computer system comprises a first memory component and a second memory component, and wherein access to the second memory component is more restricted than access to the first memory component.

23. **(Previously Presented)** The method of Claim 22, wherein the viral interrupt handler attempts to modify the second memory component.